CLAIM AMENDMENTS

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Currently Amended) A method for sharing an input device across a plurality of computing platforms, comprising:

asserting a first System Management Interrupt (SMI) at a first processor included in a first server blade;

switching an execution mode of the first processor included in the first server blade to a System Management Mode (SMM) in response to the first SMI;

in response to the first processor entering the SMM, initiating an out-of-band (OOB) communications channel between the first server blade and a second server blade, wherein initiating the OOB communications channel includes asserting a second SMI on a second processor included in the second server blade;

switching an execution mode of the second processor to the SMM in response to the second SMI;

routing input data generated at [[a]] the first server blade to [[a]] the second server blade over the OOB communications channel, said input data and said first SMI generated in response to receiving an input signal produced by an input device coupled to [[a]] the first server blade; and

providing the input data to an operating system running on the second server blade, wherein routing input data to the first and second server blades and providing the input data to the operating system are performed via a first firmware and a second firmware on the first and second server blades, respectively, wherein the first and second firmware utilize an Extensible Firmware Interface (EFI) framework ~~to perform the method for sharing an input device across the plurality of computing platforms~~.

2. (Previously Presented) The method of claim 1, wherein the method is performed via firmware in a manner that is transparent to the operating system running on the second server blade.

3. (Previously Presented) The method of claim 1, wherein the input device comprises one of a keyboard and mouse.

4. (Currently Amended) A method for sharing keyboard, video and mouse resources across a plurality of computing platforms, comprising:

asserting a first System Management Interrupt (SMI) at a first processor included in a resource host server blade;

switching an execution mode of the first processor included in the resource host server blade to a System Management Mode (SMM) in response to the first SMI;

in response to the first processor entering the SMM, initiating an out-of-band (OOB) communications channel between the resource host server blade and a target computing platform, wherein initiating the OOB communications channel includes asserting a second SMI on a second processor included in the target computing platform;

switching an execution mode of the second processor to the SMM in response to the second SMI;

routing user input data produced at [[a]] the resource host server blade in response to user inputs via a keyboard and mouse coupled to the resource host server blade to [[a]] the target computing platform over the OOB communications channel, wherein the first SMI is generated in response to the user inputs;

providing the user input data to an operating system running on the target computing platform, wherein routing user input data to the target computing platform and providing the user input data to the operating system running on the target computing platform are performed via a first firmware on the resource host server blade and a second firmware on the target computing platform, respectively, wherein the first and second firmware utilize an Extensible Firmware Interface (EFI) framework ~~to perform the method for sharing keyboard, video and mouse resources across the plurality of computing platforms~~;

routing video data produced by an operating system running on the target server blade to the resource host server blade; and

processing the video data at the resource host server blade to generate a video display signal to drive a video display coupled to the resource host server blade.

5. (Cancelled)

6. (Previously Presented) The method of claim 1, further comprising maintaining global resource mapping information identifying the resource host and the target server blades.

7. (Previously Presented) The method of claim 6, further comprising maintaining a local copy of the global resource mapping data on each of the plurality of server blades.

8. (Original) The method of claim 6, further comprising maintaining the global resource mapping data via a central global resource manager.

9. (Currently Amended) The method of claim 4, wherein the user input and video data are routed over [[an]] <u>the</u> out-of-band communication channel.

10. (Original) The method of claim 9, wherein the OOB communication channel comprises one of a system management bus, an Ethernet-based network, or a serial communication link.

11. (Previously Presented) The method of claim 4, wherein the plurality of server blades operate in a blade server environment.

12. (Previously Presented) The method of claim 4, wherein the method is performed in a manner that is transparent to operating systems running on the plurality of server blades.

13. (Cancelled)

14. (Previously Presented) An article of manufacture comprising a machine-readable medium having instructions stored thereon, which when executed on first and second server blades support sharing of keyboard, video and mouse resources coupled to the first server blade by performing operations including:

asserting a first System Management Interrupt (SMI) at a first processor included in a first server blade;

switching an execution mode of the first processor included in the first server blade to a System Management Mode (SMM) in response to the first SMI;

in response to the first processor entering the SMM, initiating an out-of-band (OOB) communications channel between the first server blade and a second server blade, wherein initiating the OOB communications channel includes asserting a second SMI on a second processor included in the second server blade;

switching an execution mode of the second processor to the SMM in response to the second SMI;

routing input data produced at the first server blade in response to user inputs via the keyboard and mouse to [[a]] the second server blade over the OOB communications channel, wherein the first SMI is generated in response to the user inputs;

providing the input data to an operating system running on the second server blade; and

routing video data produced by the operating system running on the second server blade to a video signal generation component on the first server blade, wherein routing input data produced at the first server blade to the second server blade and providing the input data to the operating system running on the second server blade are performed via a first firmware and a second firmware on the first and second server blades, respectively, wherein the first and second firmware utilize an Extensible Firmware Interface (EFI) framework ~~to perform the operations~~.

15. (Cancelled)

16. (Original) The article of manufacture of claim 14, wherein the article comprises a flash device.

17. (Previously Presented) The article of manufacture of claim 14, wherein the operations are performed in a manner that is transparent to the operating system running on the second server blade.

18. (Canceled).

19. (Previously Presented) The method of claim 1, wherein the first and second firmware runs in the pre-boot prior to operating system load.

20. (Previously Presented) The method of claim 1, wherein the first and second firmware runs during runtime of the operating system.

21. (New) A method for sharing resources across a plurality of computing platforms, the method comprising:

receiving a request for a first server blade to access a shared resource hosted by a second server blade;

using first firmware located at the first server blade, determining the second server blade via which the shared resource may be accessed, wherein the first firmware implements an Extensible Firmware Interface (EFI) framework;

entering a System Management Mode (SMM) at the first server blade and the second server blade;

in response to entering the SMM, initiating an out-of-band (OOB) communications channel between the first server blade and the second server blade;

sending the request to the second server blade from the first server blade over the OOB communications channel; and

using second firmware located at the second server blade, accessing the shared resource, wherein the second firmware implements the EFI framework.

22. (New) The method of claim 21, wherein the method is performed in a manner that is transparent to operating systems running on the plurality of computing platforms.

23. (New) The method of claim 21, wherein the OOB communication channel comprises one of a system management bus, an Ethernet-based network, or a serial communication link.

24. (New) The method of claim 21, wherein a target resource comprises a storage device.

25. (New) The method of claim 24, wherein the resource access request comprises a storage device write request, and the method further comprises sending data corresponding to the storage device write request via the OOB communication channel.

26. (New) The method of claim 24, wherein the resource access request comprises a storage device read request, and the method further comprises:

retrieving data corresponding to the read request from the shared resource; and

sending the data that are retrieved back to the first computing platform via the OOB communication channel.

27. (New) The method of claim 21, further comprising:

maintaining global resource mapping data identifying which resources are accessible via which computing platforms; and

employing the global resource mapping data to determine which computing platform to use to access the shared resource.

28. (New) The method of claim 27, wherein a local copy of the global resource mapping data is maintained on each of the plurality of computing platforms.

29. (New) The method of claim 27, wherein the global resource mapping data is maintained by a central global resource manager.

30. (New) A method for sharing a plurality of storage devices across a plurality of computing platforms, the method comprising:

configuring the plurality of storage devices as a virtual storage volume;

maintaining a global resource map that maps input/output (I/O) blocks defined for the virtual storage volume to corresponding storage devices that actually host the I/O blocks;

receiving a data access request identifying an I/O block from which data are to be accessed via the virtual storage volume wherein the request is for a first server blade to access the data;

using first firmware located at the first server blade, identifying a second server blade via which a target storage device that actually hosts the I/O block may be accessed through use of the global resource map, wherein the first firmware implements an Extensible Firmware Interface (EFI) framework;

entering a System Management Mode (SMM) at the first server blade and the second server blade;

in response to entering the SMM, initiating an out-of-band (OOB) communications channel between the first server blade and the second server blade;

routing the data access request to the second server blade that is identified, the data access request being routed from the first server blade to the second server blade over the OOB communications channel; and

using second firmware located at the second server blade, accessing the I/O block on the target storage device, wherein the second firmware implements the EFI framework.

31. (New) The method of claim 30, further comprising:

configuring the plurality of storage devices as at least one RAID (redundant array of independent disks) virtual storage volume;

maintaining RAID configuration mapping information that maps input/output (I/O) blocks defined for said at least one RAID virtual storage volume to corresponding storage devices that actually host the I/O blocks; and

employing the RAID configuration mapping information to access appropriate storage devices in response to read and write access requests.

32. (New) The method of claim 31, wherein the RAID virtual storage volume is configured in accordance with the RAID-1 standard.

33. (New) A blade server system, comprising:

a chassis, including a plurality of slots in which respective server blades may be inserted;

an interface plane having a plurality of connectors for mating with respective connectors on inserted server blades and providing communication paths between the plurality of connectors to facilitate in out of band (OOB) communication channel; and

a plurality of server blades, each including a processor and firmware executable thereon to perform operations including:

receive a resource access request from an operating system running on a requesting server blade to access a shared resource hosted by at least a second server blade selected from among the plurality of server blades;

using first firmware located at the first server blade, determining a target resource host from among the plurality of server blades that hosts a target resource that may service the resource access request, wherein the first firmware implements an Extensible Firmware Interface (EFI) framework;

entering a System Management Mode (SMM) at the first server blade and the second server blade;

in response to entering the SMM, initiating an out-of-band (OOB) communications channel between the first server blade and the second server blade;

sending the resource access request to the target resource host, the resource access request being sent from the first server blade to the second server blade over the OOB communications channel; and

using second firmware located at the second server blade, accessing the target resource via the target resource host to service the resource access request, wherein the second firmware implements the EFI framework.

34. (New) The blade server system of claim 33, wherein the operations are performed in a manner that is transparent to operating systems which are run on the plurality of server blades.

35. (New) The blade server system of claim 33, wherein each processor supports a hidden execution mode that is employed for facilitating communication via the OOB channel.